



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: )  
Takahiro ISHIKAWA, et al. )  
Serial No.: To be Assigned ) Group Art Unit: To be Assigned  
Filed: February 7, 2001 ) Examiner: To be Assigned

For: **COMPILER FOR COMPILED SOURCE PROGRAMS IN AN OBJECT-ORIENTED PROGRAMMING LANGUAGE**

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN  
APPLICATION IN ACCORDANCE  
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents  
Washington, D.C. 20231*

*Sir:*

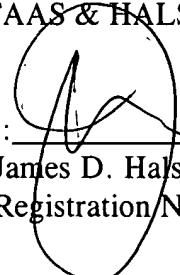
In accordance with the provisions of 37 C.F.R. §1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2000-305605  
Filed: October 5, 2000

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

By: 

James D. Halsey, Jr.  
Registration No. 22,729

700 11th Street, N.W., Ste. 500  
Washington, D.C. 20001

(202) 434-1500

Date: 7/1/01

日本国特許庁  
PATENT OFFICE  
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application: 2000年10月 5日

出願番号

Application Number: 特願2000-305605

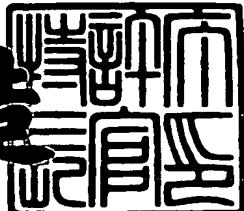
出願人

Applicant(s): 富士通株式会社

2000年12月15日

特許庁長官  
Commissioner,  
Patent Office

及川耕造



出証番号 出証特2000-3104719



**【書類名】** 特許願  
**【整理番号】** 0051508  
**【提出日】** 平成12年10月 5日  
**【あて先】** 特許庁長官殿  
**【国際特許分類】** G06F 09/45  
**【発明の名称】** オブジェクト指向型プログラミング言語によるソース・  
 プログラムをコンパイルするコンパイラ  
**【請求項の数】** 5  
**【発明者】**  
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通  
 株式会社内  
**【氏名】** 石川 貴洋  
**【発明者】**  
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通  
 株式会社内  
**【氏名】** 田中 啓士郎  
**【発明者】**  
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通  
 株式会社内  
**【氏名】** 山中 豊  
**【特許出願人】**  
**【識別番号】** 000005223  
**【氏名又は名称】** 富士通株式会社  
**【代理人】**  
**【識別番号】** 100103528  
**【弁理士】**  
**【氏名又は名称】** 原田 一男  
**【電話番号】** 045-290-2761

【手数料の表示】

【予納台帳番号】 076762

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9909129

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 オブジェクト指向型プログラミング言語によるソース・プログラムをコンパイルするコンパイラ

【特許請求の範囲】

【請求項 1】

オブジェクト指向型プログラミング言語によるソース・プログラムをコンパイルするコンパイラを格納した記録媒体であって、

前記コンパイラは、コンピュータに、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の前に前記クラスによるオブジェクトの構築命令ルーチンを呼び出す命令を生成し、記憶装置に格納する構築命令ルーチン呼出命令生成ステップと、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の後に前記クラスによるオブジェクトの消滅命令ルーチンを呼び出す命令を生成し、記憶装置に格納する消滅命令ルーチン呼出命令生成ステップと、

を実行させることを特徴とする記録媒体。

【請求項 2】

前記コンパイラは、コンピュータに、

前記ソース・プログラムから中間言の生成時に、

並列実行対象と判断される可能性のあるクラス変数を特定した場合に、当該クラスの中間言に構築・消滅命令情報域を確保するステップと、

前記構築・消滅命令情報域に、前記クラスによるオブジェクトの構築命令ルーチン及び消滅命令ルーチンに関する情報を格納するステップと、

をさらに実行させ、

前記構築命令ルーチン呼出命令生成ステップ及び前記消滅命令ルーチン呼出命令生成ステップにおいて、前記構築・消滅命令情報域に格納された情報を用いることを特徴とする請求項 1 記載の記録媒体。

【請求項 3】

前記構築・消滅命令情報域が、前記クラスの型情報域からアクセスできるようなデータ構造になっていることを特徴とする請求項2記載の記録媒体。

【請求項4】

前記コンパイラが、共有メモリ型の並列コンピュータのためのコンパイラであることを特徴とする請求項1乃至3のいずれか1記載の記録媒体。

【請求項5】

オブジェクト指向型プログラミング言語によるソース・プログラムをコンパイルするコンパイル装置であって、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の前に前記クラスによるオブジェクトの構築命令ルーチンを呼び出す命令を生成し、記憶装置に格納する手段と、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の後に前記クラスによるオブジェクトの消滅命令ルーチンを呼び出す命令を生成し、記憶装置に格納する手段と、

を有するコンパイル装置。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は、コンパイラに関し、より詳しくは、オブジェクト指向型プログラミング言語により書かれたソース・プログラムをコンパイルするコンパイラに関する。

【0002】

【従来の技術】

J a v a (Sun Microsystems社の商標) やC++などのオブジェクト指向型プログラミング言語は、プログラムの部品化等の理由により多くのプログラム開発で用いられている。また、複数のC P U (Central Processing Unit) を有する計算機も、C P U価格の下落等の理由で普及してきている。

## 【0003】

## 【発明が解決しようとする課題】

よって、オブジェクト指向型プログラミング言語で書かれたソース・プログラムから生成されたオブジェクト・プログラムを複数のC P Uを有する計算機で実行する場合も生ずるが、従来では、複数のC P Uの能力を引き出すような並列化が行われていなかった。

## 【0004】

よって、本発明の目的は、オブジェクト指向型プログラミング言語で書かれたソース・プログラムでも並列処理できるようにコンパイルすることができるコンパイラ技術を提供することである。

## 【0005】

## 【課題を解決するための手段】

並列処理を実現するには、並列化指示文で指示されたクラス変数又は並列実行対象の実行文に含まれるクラス型変数のために、ソース・プログラムにおいて指定されたオブジェクト（図1の元のオブジェクト1000）の他に、当該クラスのオブジェクト（図1の並列化処理用オブジェクト1010及び1020）を元のオブジェクト1000から生成しなければならない。また、並列処理終了時には、当該オブジェクトを消滅させなければならない。

## 【0006】

よって、オブジェクト指向型プログラミング言語によるソース・プログラムをコンパイルするコンパイラは、並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、並列実行対象の実行文又は並列化指示文により並列化される実行文の前に当該クラスによるオブジェクトの構築命令ルーチンを呼び出す命令を生成し、記憶装置に格納するステップと、並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、並列実行対象の実行文又は並列化指示文により並列化される実行文の後に当該クラスによるオブジェクトの消滅命令ルーチンを呼び出す命令を生成し、記憶装置に格納するステップとをコンピュータに実行させる。

## 【0007】

これにより必要なオブジェクトが実行時に生成されるため、並列処理が実現されるようになる。また、不要になれば生成されたオブジェクトは消滅する。

【0008】

なお、上で述べたようなコンパイラが通常のコンピュータで実行されれば当該コンピュータはコンパイル装置となる。また、コンパイラは、例えばフロッピーディスク、CD-ROM、光磁気ディスク、半導体メモリ、ハードディスク等の記憶媒体又は記憶装置に格納される。また、コンパイラの処理途中の中間的なデータは、コンピュータのメインメモリなどの記憶装置に格納される。

【0009】

【発明の実施の形態】

本発明の一実施の形態に係る機能ブロック図を図2に示す。コンピュータ100は、オブジェクト指向型プログラミング言語により記述されたソース・プログラムを格納したソースプログラム・ファイル110に対してコンパイル処理を行うコンパイラ120を実行する。コンパイラ120の処理結果は、オブジェクトコード・ファイル130として出力される。このオブジェクトコード・ファイル130は、複数のCPUを有する並列コンピュータで実行される。なお、並列コンピュータには、様々な形式のものが存在しているが、以下ではOpenMP API (Application Program Interface) を前提としたプログラムの例を示すので、本実施の形態では共有メモリ型の並列コンピュータが対象となる。

【0010】

コンパイラ120には、ソース・プログラムをコンパイラ内の処理のための中間言に変換するフロントエンド部122と、フロントエンド部122により生成された中間言に対して並列化のための処理を実施する並列化処理部124と、並列化処理部124の処理結果に基づきオブジェクトコードを生成するオブジェクトコード生成部126とが含まれる。なお、コンパイラ120においては必要に応じてコード生成の前に最適化処理などが実行される。

【0011】

次にフロントエンド部122について説明する。本実施の形態では、従来の中間言生成処理に加え図4に示すような処理を追加して、従来とは異なる中間言（

図3)を生成する。従来においては、クラス毎に、当該クラスを特定するためのクラス情報域10と、当該クラス情報域に含まれる型情報域指標12から指示される、当該クラスの型(タイプ)情報を格納する型情報域20とが設けられていた。型情報域20には、当該クラスのコンストラクタ(クラスの構築命令ルーチン)とデストラクタ(クラスの消滅命令ルーチン)の情報が格納される。本実施の形態では、OpenMPのAPIに従って記述される並列化指示文の節(c1ause)にクラスが指定されている場合にはそのクラスについて、自動並列化処理を行う場合には全てのクラスについて、以下のような情報が従来の中間言に追加される。

## 【0012】

すなわち、構築・消滅命令情報域30と、型情報域20には当該構築・消滅命令情報域30を指示する構築・消滅命令情報域指標22とを追加する。構築・消滅命令情報域30には、構築命令ルーチン40を指示する構築命令基点32と、消滅命令ルーチン42を指示する消滅命令基点34とが設けられる。

## 【0013】

以下に、具体的なプログラムを用いて説明する。

## 【表1】

```

01  struct A{
02      int mem1,mem2;
03      A() {mem1=0;mem2=1;}
04      A(A&a) {mem1=a.mem1;mem2=a.mem2;}
05      ~A();
06      add(int I) {mem1+=I;mem2+=I;}
07  } st;
08  A::~A() {printf("dtor call");}
09  main() {
10  #pragma omp parallel for private(st)
11  for (I=0;I<100000000;I++) {
12      st.add(I);

```

13 }

14 }

なお、左端の番号は以下の説明のためにのみ追加されたものであって、通常は含まれない。

#### 【0014】

上のプログラムはC++で書かれたソースプログラムであって、第1行はAという構造体を定義する部分である。第2行はmem1及びmem2という整数型のメンバを定義する文である。第3行は、コンストラクタA()を定義する文である。第4行はコピーコンストラクタである。第5行はデストラクタ～A()である。第6行はaddというメソッドを定義する文である。第7行は、上のような構造体のクラス変数stを定義する部分である。第8行はデストラクタを定義する文である。第9行はメインプログラムmainの開始部分である。第10行は、OpenMPによる並列化指示文(#pragma omp parallel)である。なお、第10行のforは以下のforループを並列化することを示しており、private(st)はスレッド毎にクラス変数stがプライベートであることを示している。第11行はI=0からI=100000000までIを1ずつインクリメントするforループを規定している。第12行は第6行のクラスstのaddメソッドが示されている。第13行はforループの終わりを、第14行はメインプログラムの終わりを規定している。

#### 【0015】

なお、表1のプログラムを並列処理するためには、元のオブジェクトの他に並列化処理用オブジェクトを生成・消滅するために、コンストラクタであるst.mem1=0及びst.mem2=1と、デストラクタであるst.～A()が並列化処理用オブジェクト毎に必要となる。そして、st.addというメソッドが各並列化処理用オブジェクトで実行される。各並列化処理用オブジェクトにおけるforループの範囲は、並列化処理用オブジェクトの数で決まる。

#### 【0016】

表1のプログラムの場合、本実施の形態では、オブジェクト情報域10でクラスstを特定する。また、型情報域20に、コンストラクタA()及びデストラ

クタ～A()の情報を含む構造体Aの情報を格納する。さらに、構築・消滅命令情報域30に、構築命令ルーチン40（コンストラクタ）であるst.mem1=0とst.mem2=1への構築命令基点32と、消滅命令ルーチン42（デストラクタ）であるst.～A()への消滅命令基点34とが格納される。

## 【0017】

以上のような情報を従来の中間言に追加するための処理フローを図4を用いて説明する。最初に、自動並列化処理を行うか否か判断する（ステップS1）。自動並列化処理を行うか否かは、例えばコンパイラ120に対するユーザの設定がなされているか否かで判断する。もし、自動並列化処理を行う場合には、現在の処理対象の文にクラス型変数が存在するか判断する（ステップS13）。クラス型である場合にはステップS7に移行する。一方、クラス型でない場合には元の処理に戻る。

## 【0018】

ステップS1で自動並列化処理を実施しない場合には、並列化指示文の使用が許されているか判断する（ステップS3）。並列化指示文は例えばOpenMPによる並列化指示文であり、並列化指示文の使用が許されているか否かは、例えばコンパイラ120に対するユーザの設定などにより判断する。もし、並列化指示文の使用が許されていない場合には、元の処理に戻る。一方、並列化指示文の使用が許可されている場合には、並列化指示文の節に指定された変数がクラス型か否かを判断する（ステップS5）。表1の例では、節はprivateで、変数はstでありクラス型である。クラス型でなければ、元の処理に戻る。

## 【0019】

もし並列化指示文の節に指定された変数がクラス型である場合には、当該クラスに対して構築・消滅命令情報域30を確保し、型情報域20内に構築・消滅命令情報域指標22を設定する（ステップS7）。そして、型情報域20から構築命令ルーチン40と、消滅命令ルーチン42とを取り出し（ステップS9）、構築・消滅命令情報域30内に、構築命令ルーチン40に対する構築命令基点32と、消滅命令ルーチン42に対する消滅命令基点34とを設定する（ステップS11）。

## 【0020】

これにより図3に示されるような中間言が生成される。図3に示されるような中間言が記憶装置に用意されていれば、次の並列化処理を高速且つ確実に行うことができるようになる。

## 【0021】

次に並列化処理部124の本実施の形態において追加される部分の処理フローを図5に示す。まず、並列化処理及び自動並列化処理において、並列化指示文の節に指定された変数がクラス型か、若しくは並列対象の実行文にクラス型変数が含まれているか判断する（ステップS21）。もし、並列化指示文の節に指定された変数がクラス型でなく且つ並列対象の実行文にクラス型変数が含まれていない場合には元の処理に戻る。一方、並列化指示文の節に指定された変数がクラス型である場合、又は並列対象の実行文にクラス型変数が含まれる場合には、クラス型変数の中間言のクラス情報域10にある型情報域指標12から型情報域20を取り出す（ステップS23）。その型情報域20の構築・消滅命令情報域指標22から、構築・消滅命令情報域30を取り出す（ステップS25）。

## 【0022】

次に構築・消滅命令情報域30から、構築命令基点32と消滅命令基点34を取り出す（ステップS27）。そして、構築命令基点32がNULL（情報なし）であるか判断する（ステップS29）。もし、NULLであればステップS33にスキップする。NULLでなければ、並列化される実行文の先頭に構築命令基点32が指す構築命令ルーチン40を呼び出す命令を生成し、記憶装置に格納する（ステップS31）。

## 【0023】

そして消滅命令基点34がNULL（情報なし）であるか判断する（ステップS33）。もし、NULLであれば元の処理に戻る。NULLでなければ、並列化される実行文の最後に消滅命令基点34が指す消滅命令ルーチン42を呼び出す命令を生成し、記憶装置に格納する（ステップS35）。そして元の処理に戻る。

## 【0024】

このようにして図5において追加で生成された命令群と通常コンパイラ120の並列化処理部124で生成される命令群とを用いて、オブジェクトコード生成部126がオブジェクトコードを生成し、オブジェクトコード・ファイル130に格納する。なお上でも説明したが、並列化処理部124の後に他の処理（例えば最適化処理）が実施され、その最適化処理後の命令群に対してオブジェクトコードを生成する場合もある。

#### 【0025】

オブジェクトコードには、図1に示したように並列化処理用オブジェクトを生成するための構築命令ルーチン42を呼び出す命令に対応するオブジェクトコードと、消滅させるための消滅命令ルーチン44を呼び出す命令に対応するオブジェクトコードとが含まれる。しかし、オブジェクトコード実行時に生成される並列化処理用オブジェクトの数は、オブジェクトコードを実行する並列コンピュータの能力などに依存するためこの時点では不明である。よって、構築命令ルーチン42を呼び出す命令及び消滅命令ルーチン44を呼び出す命令は並列化処理用オブジェクトの数分生成されるわけではない。

#### 【0026】

以上本発明の一実施の形態を説明したが、本発明はこれに限定されるものではない。例えば、図5のステップS29及びS31とステップS33及びS35とは順番を入れ替えることが可能である。また、図1のコンピュータ100がネットワークに接続されており、ソースプログラム・ファイル110が他のコンピュータから送信されてきたり、オブジェクトコード・ファイル130が他のコンピュータへ送信されるような場合もある。他のコンピュータは並列コンピュータである場合もある。上ではOpenMPを前提としたプログラムの例を示したが、その他の規格等に従ったプログラムであっても良い。

#### 【0027】

(付記1)

オブジェクト指向型プログラミング言語によるソース・プログラムをコンパイルするコンパイラを格納した記録媒体であって、

前記コンパイラは、コンピュータに、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には  
、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の  
前に前記クラスによるオブジェクトの構築命令ルーチンを呼び出す命令を生成し  
、記憶装置に格納する構築命令ルーチン呼出命令生成ステップと、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には  
、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の  
後に前記クラスによるオブジェクトの消滅命令ルーチンを呼び出す命令を生成し  
、記憶装置に格納する消滅命令ルーチン呼出命令生成ステップと、  
を実行させることを特徴とする記録媒体。

【0028】

(付記2)

前記コンパイラは、コンピュータに、  
前記ソース・プログラムから中間言の生成時に、  
並列実行対象と判断される可能性のあるクラス変数を特定した場合に、当該ク  
ラスの中間言に構築・消滅命令情報域を確保するステップと、  
前記構築・消滅命令情報域に、前記クラスによるオブジェクトの構築命令ル  
ーチン及び消滅命令ルーチンに関する情報を格納するステップと、  
をさらに実行させ、

前記構築命令ルーチン呼出命令生成ステップ及び前記消滅命令ルーチン呼出命  
令生成ステップにおいて、前記構築・消滅命令情報域に格納された情報を用いる  
ことを特徴とする付記1記載の記録媒体。

【0029】

(付記3)

前記構築・消滅命令情報域が、前記クラスの型情報からアクセスできるよう  
なデータ構造になっていることを特徴とする付記2記載の記録媒体。

【0030】

(付記4)

前記コンパイラが、共有メモリ型の並列コンピュータのためのコンパイラであ  
ることを特徴とする付記1乃至3のいずれか1記載の記録媒体。

【0031】

(付記5)

オブジェクト指向型プログラミング言語によるソース・プログラムをコンパイルするコンパイル装置であって、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の前に前記クラスによるオブジェクトの構築命令ルーチンを呼び出す命令を生成し、記憶装置に格納する手段と、

並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合には、前記並列実行対象の実行文又は前記並列化指示文により並列化される実行文の後に前記クラスによるオブジェクトの消滅命令ルーチンを呼び出す命令を生成し、記憶装置に格納する手段と、

を有するコンパイル装置。

【0032】

【発明の効果】

以上のように、オブジェクト指向型プログラミング言語で書かれたソース・プログラムでも並列処理できるようにコンパイルすることができるコンパイラ技術を提供することができた。

【図面の簡単な説明】

【図1】

本発明により生成されるオブジェクトコードの実行時における模式図である。

【図2】

本発明の一実施の形態に係るコンパイラを実行するコンピュータの機能プロック図である。

【図3】

本発明の一実施の形態に係る中間言の一例を示す図である。

【図4】

図3の中間言を生成するための処理フローを示す図である。

【図5】

図3の中間言を使用する並列化処理の処理フローを示す図である。

【符号の説明】

100 コンピュータ 110 ソースプログラム・ファイル

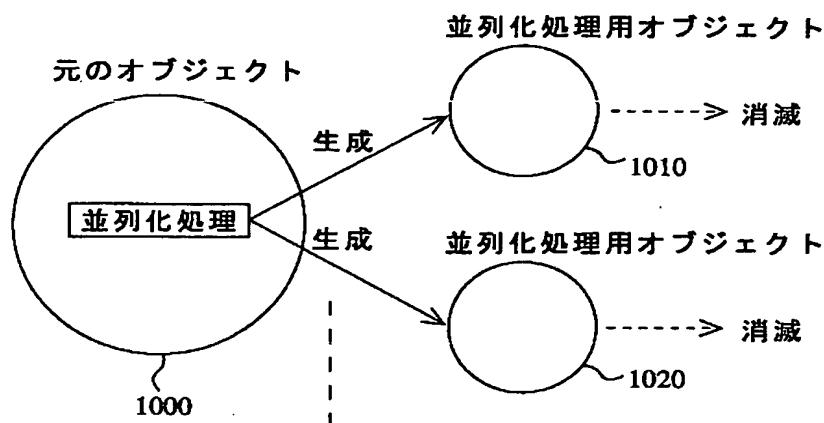
120 コンパイラ 122 フロントエンド部

124 並列化処理部 126 オブジェクトコード生成部

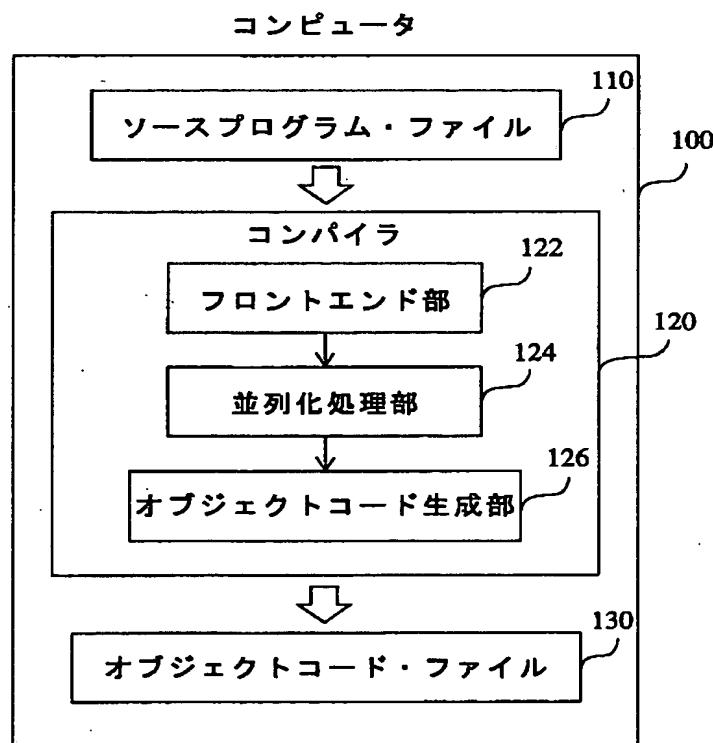
130 オブジェクトコード・ファイル

【書類名】 図面

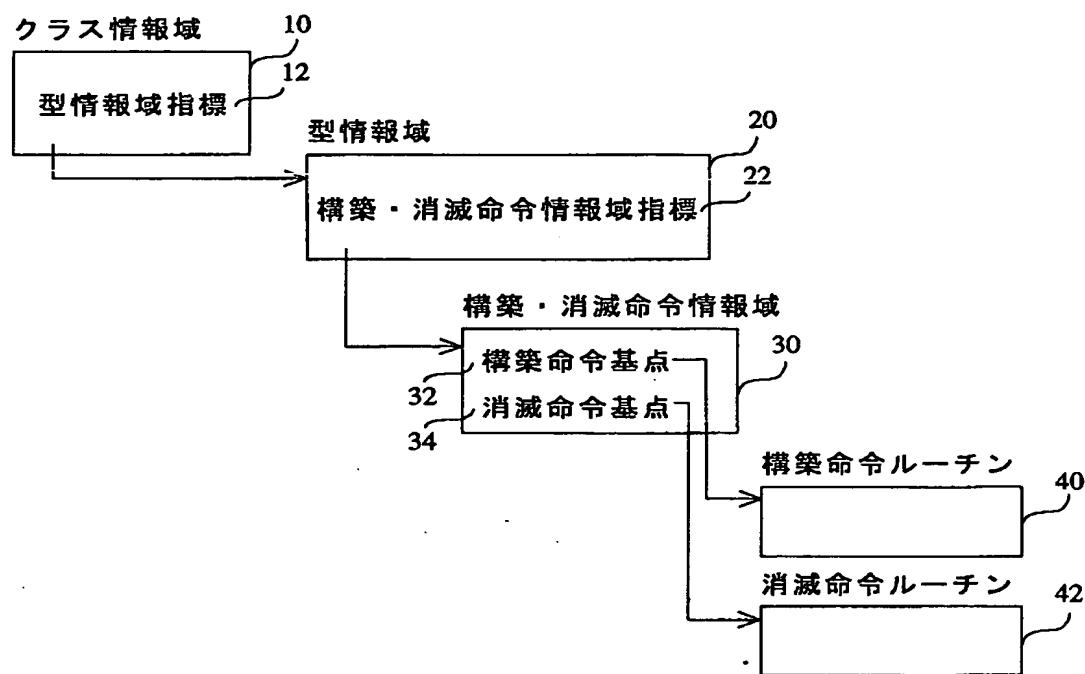
【図1】



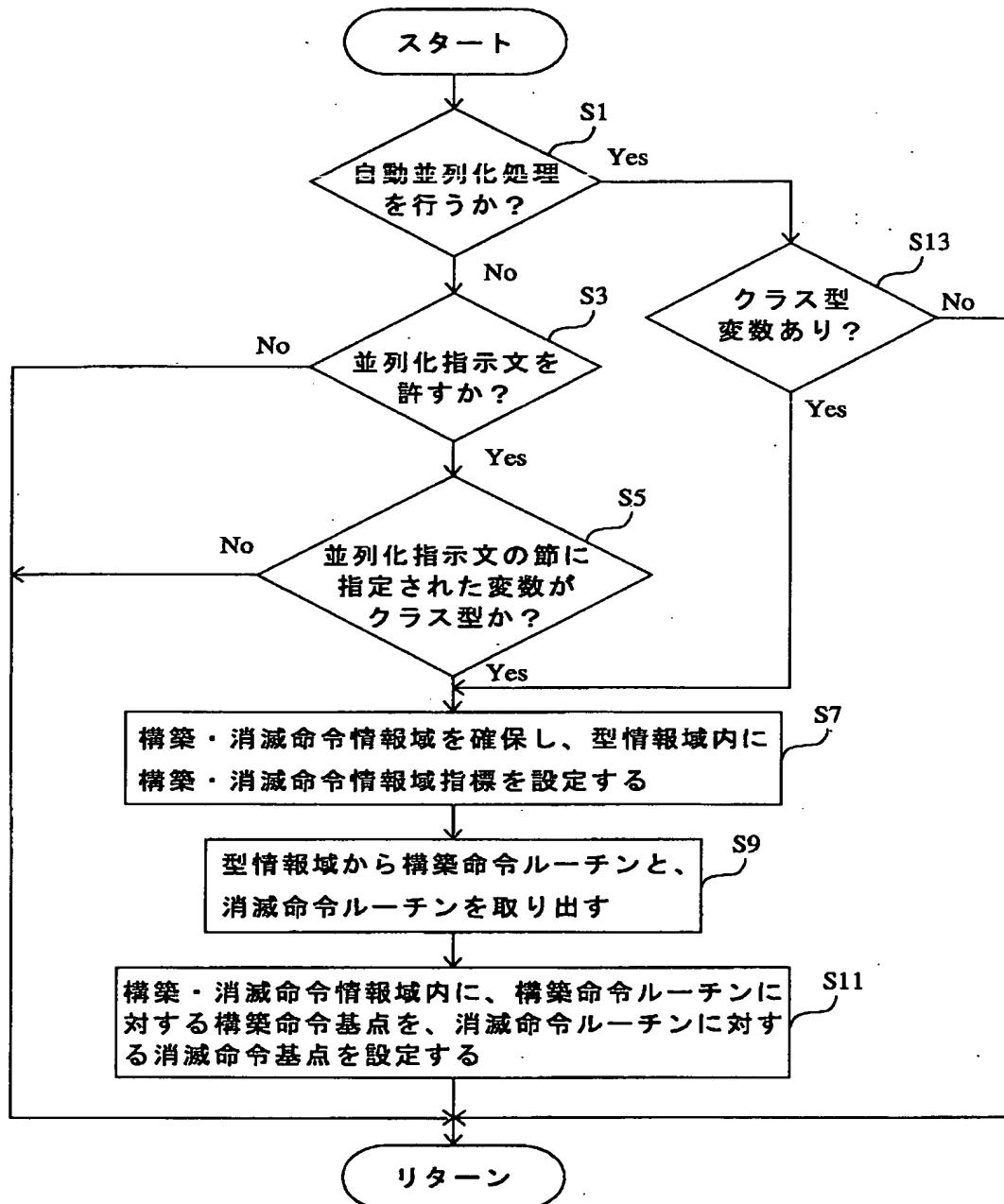
【図2】



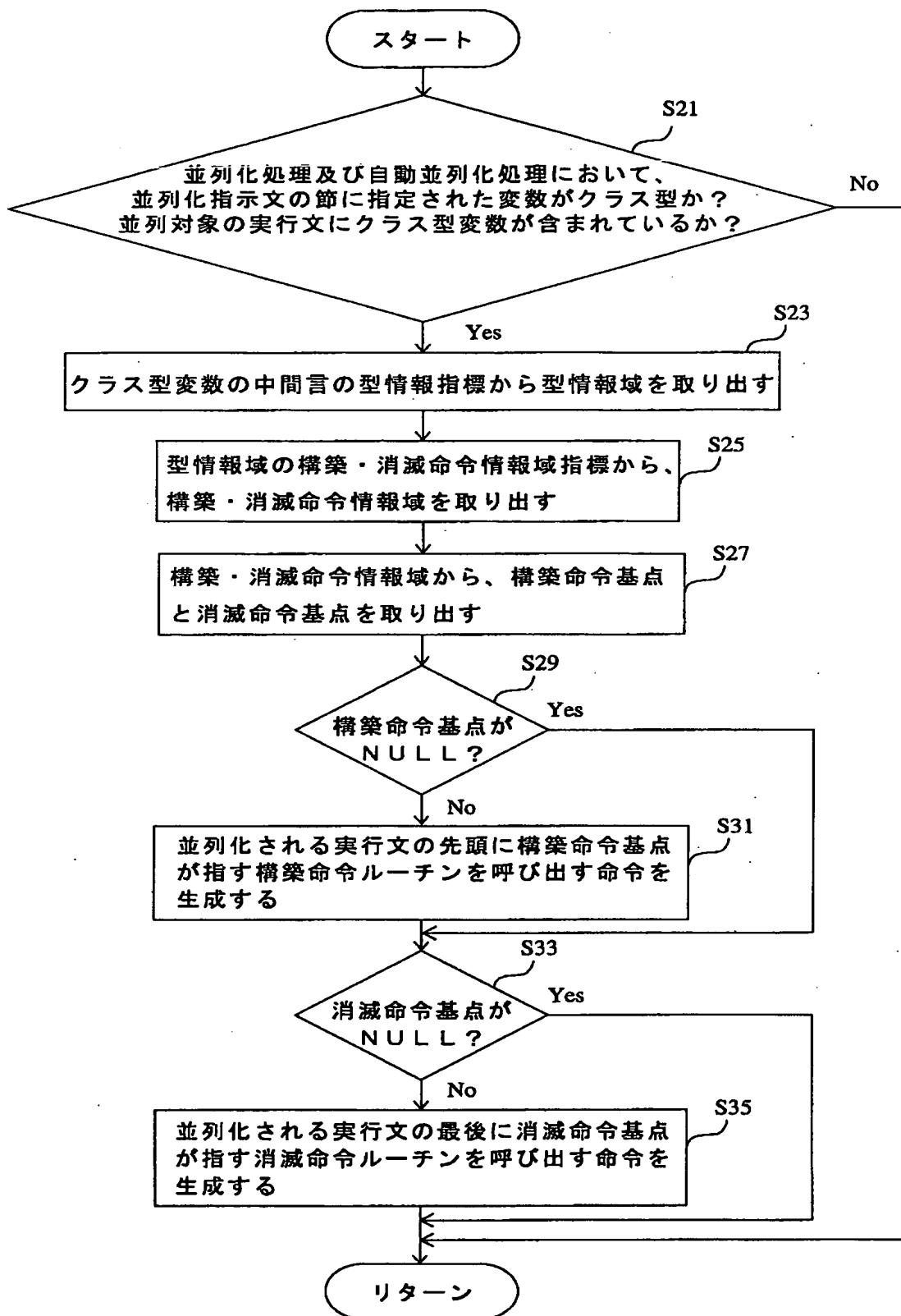
【図3】



【図4】



【図5】



【書類名】 要約書

【要約】

【課題】

オブジェクト指向型プログラミング言語によるソースプログラムでも並列処理できるようにコンパイルする。

【解決手段】

並列処理を実現するには、ソースプログラムで指定されたオブジェクト1000の他に、並列化指示文で指示されたクラス変数又は並列実行対象の実行文に含まれるクラス型変数のために、当該クラスの並列化処理用オブジェクト1010,1020を生成しなければならない。また並列処理終了時には並列化処理用オブジェクトを消滅させる。そのためコンパイラは、並列実行対象の実行文又は並列化指示文にクラス型の変数が含まれる場合に、並列実行対象の実行文又は並列化指示文により並列化される実行文の前に当該クラスによるオブジェクトの構築命令ルーチンを呼び出す命令を生成し、並列実行対象の実行文又は並列化指示文により並列化される実行文の後にそのクラスによるオブジェクトの消滅命令ルーチンを呼び出す命令を生成する。

【選択図】 図1

出願人履歴情報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号  
氏 名 富士通株式会社